Japanese Patent Laid-open No. 2000-148513 A
Publication date : May 30, 2000
Applicant : Matsushita Denki K.K.
Title : Task control method and task control device

[0005] The present invention has been achieved in view of the problems inherent in the conventional art, and an object thereof is to provide a task control method and a task control device capable of naturally assigning a low priority task to a CPU to improve through-put of a real time OS even when a high priority task continues to dominate the CPU.

[0047] Firstly, the timer-value control unit 103 receives a 1-time slice signal from the system clock 111 and counts down the timer value which is task information of each task on the executable queue 141 (step 301). As long as the timer value indicates a specific value such as "-1" as described above, the count is not decremented. Subsequently, it is determined whether a task of which timer value is "0" exists. When there is no such task, the process returns to a state waiting for the reception of the 1-time slice signal (step 301) (step 311). When the task of which timer value is "0" exists, the priority of the task is changed to the time out priority (step 321), the executed-task switch operation shown in Fig. 4 is performed, and the process returns to a state waiting for the reception of the 1-time slice signal (step 301).

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2000－148513
（P2000－148513A）
(43)公開日　平成12年５月30日(2000.5.30)

(51)Int.Cl.⁷	識別記号	ＦＩ	テーマコード（参考）
Ｇ０６Ｆ　9/46	３４０	Ｇ０６Ｆ　9/46	３４０Ｂ　５Ｂ０９８

審査請求　有　　請求項の数８　ＯＬ　（全 10 頁）

(21)出願番号　特願平10－320759

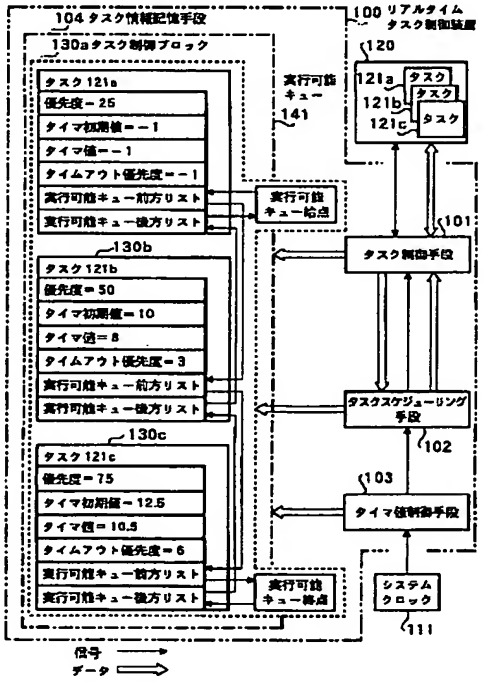
(22)出願日　平成10年11月11日(1998.11.11)

(71)出願人　000005821
松下電器産業株式会社
大阪府門真市大字門真1006番地
(72)発明者　森久　謙二郎
神奈川県横浜市港北区網島東四丁目３番１号　松下通信工業株式会社内
(74)代理人　100105050
弁理士　鷲田　公一

Ｆターム(参考)　5B098 CC03

(54)【発明の名称】　タスク制御方法およびタスク制御装置

(57)【要約】
【課題】　高優先度タスクがＣＰＵを独占し続けている場合でも、低優先度タスクを効率的にＣＰＵに割り当て、システム全体のスループットを向上させること。
【解決手段】　タスク制御ブロック１３０ａ～１３０ｃには、パラメータとして、初期優先度，タイマ初期値，タイムアウト優先度が登録される。タイムアウト優先度は初期優先度よりも高優先度である。タイマ値制御手段１０３は、タイマ値をカウントダウンしていき、タイムアップすると、タスクの優先度を初期優先度からタイムアウト優先度に変更する。タスクスケジューリング手段１０２は、変更された優先度にしたがって実行可能キュー１４１のソーティングを行い、タスク制御手段１０１は、そのソーティング順に実行するべきタスクの切換を行う。



1

【特許請求の範囲】

【請求項１】　実時間処理を必要とする複数のタスクの各々に優先度を付与し、その優先度に基づき優先判定を行ってタスクを実行する場合に、前記複数のタスクのうちの少なくとも一部のタスクについて、所定期間内に実行されなかった場合にはそのタスクの優先度を高める優先度変更処理を施すことを特徴とするタスク制御方法。

【請求項２】　あらかじめタスクに、初期優先度と、この初期優先度が適用される期間と、前記初期優先度よりも高位のタイムアウト優先度とを設定しておき、前記初期優先度が適用される期間が経過すると、前記タイムアウト優先度により優先判定を行う請求項１記載のタスク制御方法。

【請求項３】　実時間処理を必要とする複数のタスクの各々に優先度を付与し、その優先度に基づき優先判定を行ってタスクを実行するタスク制御方法において、タスクが実行可能状態になると、そのタスクについて初期優先度，タイマ初期値およびタイムアウト優先度を取得して実行可能キューに登録し、時間の経過と共にタイマ値を前記タイマ初期値よりカウントダウンしていき、前記タイマ値が「０」になるとそのタスクの優先度を前記初期優先度から前記タイムアウト優先度に変更して前記実行可能キューのソーティングを行い、前記実行可能キューのソーティングの結果として実行可能キューの先頭タスクが前回のソーティング時における先頭タスクと異なるものとなった場合、前記実行可能キューの前記先頭タスクを実行タスクとしてＣＰＵに割り当てることを特徴とするタスク制御方法。

【請求項４】　タスク生成時に、前記初期優先度，タイマ初期値，タイムアウト優先度の値を設定することを特徴とする請求項３記載のタスク制御方法。

【請求項５】　前記タイムアウト優先度の範囲が前記初期優先度の取り得る値の範囲外であり、かつ、前記タイムアウト優先度の方が前記初期優先度よりも高位であることを特徴とする請求項３記載のタスク制御方法。

【請求項６】　前記実行可能キューに登録されているタスクのタイマ値が所定の値の場合、前記タイマ値のカウントダウンが行われず、これによってそのタスクの優先度は、常に前記初期優先度となることを特徴とする請求項３記載のタスク制御方法。

【請求項７】　タスクの状態変化の検出および実行タスクの切換を行うタスク制御手段と、このタスク制御手段によって状態変化が検出されたタスクに関する、初期優先度，タイマ初期値およびタイムアウト優先度を含む情報を記憶するタスク情報記憶手段と、実行可能キューに登録されているタスクを高優先度順にソーティングするタスクスケジューリング手段と、前記実行可能キューに登録されているタスクに関するタイマ値を前記タイマ初期値からカウントダウンしていき、前記タイマ値が「０」となった場合に、そのタスクの優先度を前記初期

2

優先度からタイムアウト優先度に変更すると共に、前記タスクスケジューリング手段に実行可能キューのソーティングを要求するタイマ値制御手段と、を具備することを特徴とするタスク制御装置。

【請求項８】　前記タスクスケジューリング手段は、実行可能キューのソーティング時に優先度の同じタスクが複数存在する場合、タスクの優先度を初期優先度からタイムアウト優先度に変更したタスクをキューの高位、あるいはキューの低位にソーティングすることを特徴とする請求項７記載のタスク制御装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明はタスク制御方法およびタスク制御装置に関し、特に、複数の実時間処理を必要とするタスクの実行を制御するリアルタイムオペレーティングシステム（以下リアルタイムＯＳと記述する）を用いたタスク制御方法およびタスク制御装置に関する。

【０００２】

【従来の技術】タスクスケジューリング方法の典型的なものは、タスクに優先度を設定しておき、タスクがＣＰＵに割り当てられるのを待っている状態（以下、実行可能状態という）のときに、優先度の高位のものから順に実行可能キューに登録（ソーティング）し、実行可能キューの先頭タスクから順に実行するという方法である。

【０００３】

【発明が解決しようとする課題】しかし、従来のスケジューリング方法においては、高優先度の特定のタスクが頻発した場合や、複数のタスク実行が重なった場合などには、低優先度のタスクは、いつ実行されるか全く見当がつかず、最悪の場合には、全く実行されない、という問題を有していた。

【０００４】この問題を解決する方法としては、割り込みを利用して、低優先度のタスクを強制的にＣＰＵに割り当てるという手法が考えられる。つまり、タスクが実行可能状態に変化したら割り込みタイマをスタートさせ、設定した時間が経過するとアプリケーションレベルで割り込みをかけてタスクを実行する、というものである。しかし、この方法では高優先度の特定のタスクが頻発していない場合や、複数のタスクの実行が重なっていない場合でも、一律に割り込みタイマの設定時間の経過後でなければタスク実行がされず、結果としてシステム全体のスループットを低下させる恐れがある。

【０００５】本発明は、このような従来技術の問題点を解決するためになされたものであり、高優先度タスクがＣＰＵを占有し続けている場合でも、低優先度タスクをＣＰＵに無理なく割り当ててリアルタイムＯＳのスループットを向上させることができるタスク制御方法およびタスク制御装置を提供することを目的とする。

【０００６】

【課題を解決するための手段】本発明のタスク制御方法では、あらかじめアプリケーションが設定した時間をしきい値とし、そのしきい値の時間が過ぎると、タスクの優先度を高位の優先度に変化させ、そのタスクがＣＰＵに割り当てられる確率を高める。

【０００７】これにより、高優先度のタスクがＣＰＵを占有し続けている場合であっても、低優先度のタスクが実行される可能性が格段に高まる。しかも、アプリケーションが設定した時間が過ぎると優先度がダイナミックに変化することから、タスクが実行可能状態に変化してから実際にＣＰＵに割り当てられるまでの時間をアプリケーションが設定した値程度に保証することが可能となる。また、特別なパラメータや割り込み等の他の特別な処理を用いるわけではないので、処理が複雑化しない。すなわち、リアルタイムＯＳがサポートしている優先度順に処理するという基本的な機能をそのまま利用して低優先度のタスクの処理確率を高める、という手法をとるので、無理がなく、しかも、設定するパラメータは、時間しきい値と、しきい値の前後における優先度だけでよいため、処理手順や装置構成が複雑化しない。したがって、システムのスループットの向上を、無理なく、簡素化された方法で実現することができる。

【０００８】また、本発明のタスク制御装置は、実行可能キューに登録されているタスクを優先度順にソーティングするタスクスケジューリング手段と、実行可能キューに登録されているタスクについてタイムアウトを検出すると優先度を高位に変更すると共に、タスクスケジューリング手段に実行可能キューのソーティングを要求するタイマ値制御手段と、を有する。

【０００９】タイマ値制御手段とタスクスケジューリング手段の働きにより、低優先度のタスクについても、時間経過に伴なうダイナミックな優先度の変更が可能となり、リアルタイムＯＳがサポートするシステム全体としてのスループットが、無理なく向上する。

【００１０】

【発明の実施の形態】本発明のタスク制御方法の第１の態様では、実時間処理を必要とする複数のタスクの各々に優先度を付与し、その優先度に基づき優先判定を行ってタスクを実行する場合に、前記複数のタスクのうちの少なくとも一部のタスクについて、所定期間内に実行されなかった場合にはそのタスクの優先度を高める優先度変更処理を施す。

【００１１】これにより、低優先度のタスクが予め設定した時間程度で実行される可能性が高まり、無理なくシステム全体のスループットを高めることができる。

【００１２】また、本発明のタスク制御方法の第２の態様では、第１の態様において、あらかじめタスクに、初期優先度と、この初期優先度が適用される期間と、前記初期優先度よりも高位のタイムアウト優先度とを設定しておき、前記初期優先度が適用される期間が経過する

と、前記タイムアウト優先度により優先判定を行うようにした。

【００１３】したがって、簡単なパラメータの設定によってタスクの優先度変更処理を実現できる。

【００１４】本発明のタスク制御方法の第３の態様では、実時間処理を必要とする複数のタスクの各々に優先度を付与し、その優先度に基づき優先判定を行ってタスクを実行するタスク制御方法において、タスクが実行可能状態になると、そのタスクについて初期優先度，タイマ初期値およびタイムアウト優先度を取得して実行可能キューに登録し、時間の経過と共にタイマ値を前記タイマ初期値よりカウントダウンしていき、前記タイマ値が「０」になるとそのタスクの優先度を前記初期優先度から前記タイムアウト優先度に変更して前記実行可能キューのソーティングを行い、前記実行可能キューのソーティングの結果として実行可能キューの先頭タスクが前回のソーティング時における先頭タスクと異なるものとなった場合、前記実行可能キューの前記先頭タスクを実行タスクとしてＣＰＵに割り当てるようにした。

【００１５】これにより、高優先度タスクがＣＰＵを占有し続けている場合でも、低優先度タスクをＣＰＵに割り当てることができ、また、低優先度タスクが実行可能状態に変化してから実際にＣＰＵに割り当てられるまでの時間をアプリケーションが設定した値程度に保証することができる。

【００１６】本発明のタスク制御方法の第４の態様は、タスク生成時に、前記初期優先度，タイマ初期値，タイムアウト優先度の値を設定する。

【００１７】これにより、アプリケーションレベルで簡単にタスク制御を行える。

【００１８】本発明のタスク制御方法の第５の態様では、前記タイムアウト優先度の範囲が前記初期優先度の取り得る値の範囲外であり、かつ、前記タイムアウト優先度を前記初期優先度よりも高位とした。

【００１９】これにより、タイムアウトして待ち時間が長いタスクが、タイムアウトしていない他のタスクよりも優先的に実行される。

【００２０】また、本発明のタスク制御方法の第６の態様では、前記実行可能キューに登録されているタスクのタイマ値が所定の値の場合、前記タイマ値のカウントダウンが行われず、これによってそのタスクの優先度は、常に前記初期優先度となるようにした。

【００２１】これにより、タスク生成時に、アプリケーションが各タスクに対して個別にタイムアウト時の優先度変更機能を使用するかしないかを適宜、選択することができる。

【００２２】また、本発明のタスク制御装置にかかる第７の態様では、タスクの状態変化の検出および実行タスクの切換を行うタスク制御手段と、このタスク制御手段によって状態変化が検出されたタスクに関する、初期優

先度，タイマ初期値およびタイムアウト優先度を含む情報を記憶するタスク情報記憶手段と、実行可能キューに登録されているタスクを高優先度順にソーティングするタスクスケジューリング手段と、前記実行可能キューに登録されているタスクに関するタイマ値を前記タイマ初期値からカウントダウンしていき、前記タイマ値が「０」となった場合に、そのタスクの優先度を前記初期優先度からタイムアウト優先度に変更すると共に、前記タスクスケジューリング手段に実行可能キューのソーティングを要求するタイマ値制御手段と、を設ける構成とした。

【００２３】これにより、タイマ値の制御による優先度の変更を行う手段と、優先度にしたがった実行可能キューのソーティングを行う手段とをもつ簡素化された構成でもって、高優先度タスクのみならず、低優先度タスクにも、ほぼ所望の範囲でＣＰＵへの割り当てのチャンスが与えられるシステムを構築できる。

【００２４】また、本発明のタスク制御装置にかかる第８の態様では、第７の態様において、前記タスクスケジューリング手段は、実行可能キューのソーティング時に優先度の同じタスクが複数存在する場合、タスクの優先度を初期優先度からタイムアウト優先度に変更したタスクをキューの高位、あるいはキューの低位にソーティングする構成とした。

【００２５】これにより、優先度が同じタスクが競合した場合でも、不都合なくタスクの処理を続行することができる。

【００２６】以下、本発明の実施の形態について図面を参照して具体的に説明する。

【００２７】（発明の実施の形態）図１はリアルタイムタスクの制御を行うタスク制御装置のブロック図である。

【００２８】本実施の形態にかかるリアルタイムタスク制御装置１００は、例えばタスク１２１ａ、タスク１２１ｂおよびタスク１２１ｃ等から構成されるタスク群１２０を制御するものである。

【００２９】このリアルタイムタスク制御装置１００は、タスク制御手段１０１と、タスクスケジューリング手段１０２と、タイマ値制御手段１０３と、タスク情報記憶手段１０４とを有する。

【００３０】タスク制御手段１０１は、制御対象であるタスク群１２０上の各タスクの状態変化の検出およびタスクスイッチ（実行するタスクの切換）等のタスク制御を実行する。

【００３１】タスク情報記憶手段は、タスク制御手段１０１により検出されたタスクの状態に関する情報（少なくともタスク状態、優先度、タイマ初期値およびタイムアウト優先度を含む情報）を、各タスクに対応して設けられたタスク制御ブロックに記憶する。すなわち、タスク１２１ａ，１２１ｂ，１２１ｃに関するタスク情報は

それぞれ、タスク制御ブロック１３０ａ，１３０ｂ，１３０ｃに記憶される。ここで、タスク制御ブロック１３０ａ～１３０ｃに必要な情報が記憶されるということは、実行可能状態となっている全部のタスクが把握され、しかも、どのタスクから順に実行するべきかが、各タスクの優先度を比較することによって一義的に定まることを意味する。本明細書では、優先度にしたがって順序づけられた、実行可能なタスクに関する情報を実行可能キュー１４１として把握する。つまり、タスク情報記憶手段１０４は、実行可能キュー１４１を記憶する手段として位置づけられることになる。

【００３２】タスクスケジューリング手段１０２は、タスク制御手段１０１によって状態変化が検出されたタスクについて、優先度，タイマ値およびタイムアウト優先度を含むタスク情報を取得し、実行可能キュー１４１（つまり、タスク制御ブロック１３０ａ～１３０ｃ）に登録し、かつ、高優先度順にソーティング（並び替え）する働きをする。

【００３３】図１の装置において、高優先度の順にソーティング（並び替え）することは、タスク制御ブロック１３０ａ～１３０ｃの各々の実行可能キュー前方リストに、そのタスクよりも一つ下位の優先度をもつ次のタスクの情報の格納アドレスを書き込むこと、あるいは、実行可能キュー後方リストに、そのタスクより一つ上位の優先度をもつ次のタスクの情報の格納アドレスを書き込むこと、を意味する。つまり、タスク１３０ａ～１３０ｃの現在の優先度はそれぞれ、「２５」，「５０」，「７５」であり、タスク１３０ａが最優先のタスクであり、次に優先されるのがタスク１３０ｂであり、次がタスク１３０ｃである。したがって、実行可能キューのソーティングがなされると、図１に矢印で示されるような、優先度にしたがった順序付けがさなれることになる。

【００３４】タイマ値制御手段１０３は、システムクロック生成手段１１１から供給されるシステムクロックを用いてタイムスライス毎に（つまり、システムクロックの入力タイミング毎に）、実行可能キュー１４１内の各タスク制御ブロックのタイマ値を、タイマ初期値からカウントダウンしていく。そして、タイマ値が「０」となった場合、そのタスクの優先度をタイムアウト優先度に変更すると共に、タスクスケジューリング手段１０２に、実行可能キュー１４１のソーティング（並び替え）を要求する。

【００３５】なお、図１において、タスク１２１ａに対応するタスク制御ブロック１３０ａのタイマ初期値（およびタイマ値）が「－１」となっているが、この「－１」が設定されると、タイマ値のカウントダウンによる優先度変更処理が行われない。つまり、アプリケーションがタスクの優先度を固定したいと判断した場合には、タイマ初期値に「－１」をセットすることによって、簡

単に、優先度変更処理を禁止することができる。

【００３６】以上のように構成されたリアルタイムタスク制御装置１００の動作とその効果について、図２～図８を参照して具体的に説明する。なお、図９～図１２は、比較例（従来例）の動作と効果を示す図である。

【００３７】リアルタイムタスクの制御動作は、大きく分けて図２に示すタスクスケジューリング動作と、図３に示すタイマ値制御動作と、図４に示す実行タスクスイッチ動作の３つから構成される。なお、実行タスクスイッチ動作は、タスクスケジューリング動作およびタイマ値制御動作の一連の手順に包含されるものである。また、タイマ値制御動作は、本発明における核となる部分である。

【００３８】図１に示すように、実行可能状態のタスクは、タスク１２１ａ～１２１ｃの３つである。そして、アプリケーションは、タスク生成時にパラメータとして少なくとも優先度，タイマ初期値，タイムアウト優先度の３つの値を設定する。優先度の設定に関し、例えば、アプリケーションが設定するタイムアウト優先度の範囲を０から９とし、同じくアプリケーションが設定する初期優先度を１０から９９として、初期優先度とタイムアウト優先度の範囲が重ならないように別々に設けることもできる。なお、本実施の形態では、優先度の数値が低いほど優先順位が高い。

【００３９】まず、タスクスケジューリング動作について説明する。

【００４０】図２に示すように、まず、タスク制御手段１０１が、制御対象であるタスク群１２０上の各タスクに対してタスク状態の変化を検出する（ステップ２０１）。そして、検出したタスク状態変化に対してタスク生成，タスク削除またはそれ以外の３つの場合分けを行う（ステップ２１１）。

【００４１】タスクが新たに生成された場合は、タスク情報記憶手段１０４が、生成されたタスクに対するタスク制御ブロック１３０を確保し（ステップ２１２）、少なくともタスク状態，優先度，タイマ初期値およびタイムアウト優先度を含むタスク情報をタスク制御ブロック１３０に書き込んで制御ブロックの編集を行う（ステップ２１３）。

【００４２】一方、タスクが削除された場合は、タスク情報記憶手段１０４が、削除されたタスクに対するタスク制御ブロック１３０を解放する（ステップ２１４）。それ以外のタスク状態変化が起こった場合は、タスク情報記憶手段１０４が、状態変化したタスクに対するタスク制御ブロック１３０のタスク状態を変更してブロックの編集を行う（ステップ２１３）。

【００４３】そして、タスク制御手段１０１は、検出したタスク状態変化を、実行不可能状態から可能状態への変化，実行可能状態から不可能状態への変化，あるいはそれ以外の変化の３つの場合に分ける（ステップ２２１）。ここで、タスクが実行不可能状態であるとは、タスクが休止状態であるか、遅延状態であるか、または事象待ち状態であるかのいずれかの状態であることをいう。

【００４４】タスクが実行不可能状態から可能状態に変化した場合は、タスク制御手段１０１がタスク情報記憶手段１０４から状態変化したタスクに対するタスク制御ブロック１３０の優先度、タイマ初期値及びタイムアウト優先度を取得し（ステップ２２２）、タスクスケジューリング手段１０２が、この取得したタスク情報と共にタスクを実行可能キュー１４１に追加し（ステップ２２３）、図４に示す実行タスクスイッチ動作を行い、タスク制御手段１０１のタスク状態変化検出（ステップ２０１）待ち状態に戻る。タスクを実行可能キュー１４１に追加する際、タイマ初期値はタイマ値に格納する。

【００４５】タスクが実行可能状態から不可能状態に変化した場合は、タスクスケジューリング手段１０２が、実行可能キュー１４１からそのタスクの情報を削除し（ステップ２２４）、図４に示す実行タスクスイッチ動作を行い、タスク制御手段１０１のタスク状態変化検出（ステップ２０１）待ち状態に戻る。それ以外のタスク状態変化が起こった場合は、タスク制御手段１０１のタスク状態変化検出（ステップ２０１）待ち状態に戻る。

【００４６】次に、図３を用いて、タイマ値制御動作を説明する。

【００４７】まず、タイマ値制御手段１０３が、システムクロック１１１から１タイムスライス信号を受信し、実行可能キュー１４１上の各タスクのタスク情報であるタイマ値をカウントダウンする（ステップ３０１）。なお、上述のようにタイマ値が「－１」等特定の値となっていると、カウントダウンは行われない。続いてタイマ値が「０」であるタスクが存在するかどうかを判定し、存在しない場合は１タイムスライス信号受信（ステップ３０１）待ち状態に戻る（ステップ３１１）。タイマ値が「０」であるタスクが存在する場合は、そのタスクの優先度をタイムアウト優先度に変更し（ステップ３２１）、図４に示す実行タスクスイッチ動作を行い、１タイムスライス信号受信（ステップ３０１）待ち状態に戻る。

【００４８】次に、図４を用いて、実行タスクスイッチ動作について説明する。

【００４９】まず、タスクスケジューリング手段１０２が、実行可能キュー１４１上のタスクを優先度の高い順にソーティングする（ステップ４０１）。すなわち、図１の各タスク制御ブロック１３０ａ～１３０ｂ内の実行可能キュー前方リスト，後方リストのアドレスを書き込んで、各タスクの優先順序を決定する。

【００５０】高優先度順にタスクをソーティングする際、優先度の同じタスクが複数存在する場合、タスクが実行可能キューに追加された順にキューの高位にソーテ

ィングする方法の他、優先度をタイムアウト優先度に変更した方のタスクをキューの高位にソーティングする方法や、優先度をタイムアウト優先度に変更した方のタスクをキューの低位にソーティングする方法を採用することにより、同じ優先度のタスクが競合する場合にも不都合なくソーティングを行える。

【００５１】続いて、タスクスケジューリング記憶手段１０５は、実行可能キューの先頭タスク（最高優先度タスク）が変化したかどうかを検出し（ステップ４１１）、変化しなかった場合は実行タスクスイッチ動作を終了する。変化した場合は、タスクスケジューリング記憶手段１０５は、タスク制御手段１０１に実行タスクスイッチ要求信号を送信し、実行タスクスイッチ要求信号を受信したタスク制御手段１０１は、タスク情報記憶手段１０４を用いてタスク制御ブロックの書き換えおよびタスクスイッチの実行時に必要なタスク情報の取得を行い（ステップ４２１）、タスク群１２０に対して実行タスクのスイッチを行い（ステップ４２２）、実行タスクスイッチ動作を終了する。

【００５２】このようにして、アプリケーションが設定した所定の時間がタイムアウトすると、低優先度のタスクの優先度がステップ的に変更されて、そのタスクの実行確率が高まる。図１では、タスク１２１ａが最も優先度が高く，続いてタスク１２１ｂ，１２１ｃの順となっていた。ここで、タイムアウトに伴う優先度変更によって、タスク１２１ｂの優先度がもっとも高位になり、続いてタスク１２１ｃ、タスク１２１ａの順になると、実行可能キュー１４１のソーティング後の状態は、図５のように変化する。このように、所定の時間経過と共にタスクの優先度をダイナミックに変更し、その都度、実行可能キューのソーティングを行うことで、低い優先度のタスクも、アプリケーションが設定した時間程度で実行されることになる。

【００５３】本発明の実施の形態によるタスク実行の様子を、図６～図８に示す。

【００５４】図６は、３つの周期タスク（タスク１２１ａ、タスク１２１ｂ、タスク１２１ｃ）の、１周期内におけるタスク実行時間およびタスクの実行周期を示す図である。図７は、図６に示すような３つの周期タスクに対して、図１のタスク制御ブロック１３０ａ～１３０ｃに示される値の優先度，タイマ初期値，タイムアウト優先度を付与して（つまり、タスク１２１ａに対しては優先度２５，タイマ初期値－１，タイムアウト優先度-1を付与し、タスク１２１ｂに対しては優先度５０，タイマ初期値１０，タイムアウト優先度３を付与し、タスク１２１ｃに対しては、優先度７５，タイマ初期値１２．５，タイムアウト優先度６を付与して）、本発明の実施の形態によるタスク実行を行った場合の時間軸に対する各タスクの優先度の変化および実行タスクの変化を示す図である。図８は、図７で示された実行タスクの変化から各タスクのタスク実行待ち時間の最高値を最高タスク実行待ち時間として、合計値をタスク実行待ち時間合計として示すものである。

【００５５】図７において、白抜きの三角印はタスクが実行状態に変化したことを示し、黒い三角印は、１周期分の処理を終了して実行不可能状態に変化したことを示す。また、図７の下側の図において、実線の横線はタスクが実行されている期間を示し、点線の横線はタスクが実行待ち状態となっている期間を示している。

【００５６】図７の上側の図に示されるように、１０タイムスライス（１０ＴＳ）が経過するとタイムアウトによりタスク１２１ａの優先度がもっとも高くなり、１２．５タイムスライス（１２．５ＴＳ）が経過すると、タイムアウトに伴いタスク１２１ｂの優先度がタスク１２１ａに続いて高位となる。これにより、図７の下側に示すように、初期優先度が低くかったタスク１２１ｂ，１２１ｃも、アプリケーションが設定したタイマ初期値（タスク１２１ｂについては１０ＴＳ、タスク１２１ｃについては１２．５ＴＳ）程度の待ち時間で実行される。

【００５７】図８の左側に示されるように、タスク１２１ａの待ち時間は６．２５ＴＳであり、タスク１２１ｂの待ち時間は１０ＴＳであり、タスク１２１ｃの待ち時間は１５ＴＳである。このデータから、各タスクがＣＰＵに効果的に割り当てられ、システム全体としての処理効率が向上しているのがわかる。

【００５８】図９および図１０は、優先度の変更を行わない従来方式によるタスク実行の様子とその結果を比較例として示すものである。

【００５９】図９から明らかなように、低優先度のタスクは上位のタスクの処理が終わるまで、待たされ続けることになる。したがって、図１０の左側に示されるように、最も上位のタスク１２１ａは待ち時間が０であるものの、タスク１２１ｂについては待ち時間が１５，タスク１２１ｃについては待ち時間が２５となり、図８の本実施の形態の場合と比べ、低優先度のタスクの待ち時間がかなり長い。また、図８と図１０の比較から、最高タスク実行待ち時間値およびタスク実行待ち時間の合計値についても、本実施の形態のほうがトータルの値が小さく、効率的な処理がなされていることがわかる。

【００６０】図１１および図１２に、アプリケーションレベルで割り込みタイマを使用した従来例のタスク実行の様子およびその結果を示す。

【００６１】割り込みを使用して低優先度のタスクを強制的に実行させる従来例では、ＣＰＵが空き状態でも、割り込みが実行されるまで実行タスクの切り替えが行われることはなく、結果としてシステムのスループットを低下させるという不合理な結果が生じることがある。これに対し、本実施の形態によるタスク実行ではＣＰＵが空き状態になればタイマ値が０でなくともＣＰＵへの割

り当てが行われるので、結果としてシステムのスループットを低下させることはない。

【００６２】図１１においては、時刻２１．２５ＴＳでＣＰＵが空き状態であるにも関わらずタスク１２１ｂは実行されないが、図７では、タスク１２１ｂは２１．２５ＴＳで実行される。

【００６３】図８に示されるように、本実施の形態では、タスク１２１ｂおよびタスク１２１ｃの最高タスク実行待ち時間はタイマ初期値程度に抑えられている。また、タスク１２１ａのタスク実行待ち時間合計も、アプリケーションレベルで割り込みタイマを使用する場合（図１２）よりも小さく抑えられていることがわかる。

【００６４】

【発明の効果】以上説明したように本発明によれば、リアルタイムタスク制御装置にタイムアウト時の優先度変更機能を設けることにより、高優先度タスクがＣＰＵを占有し続けている場合でも、低優先度タスクをシステム全体のスループットを低下させること無くＣＰＵに割り当て、また、低優先度タスクが実行可能状態に変化してから実際にＣＰＵに割り当てられるまでの時間をアプリケーションが設定した値程度に保証することができる。これにより、システム全体のスループットが向上する。また、時間に関する制御は、システムクロックから得られるタイムスライス毎にタイマ値をカウントダウンする処理のみなので、リアルタイムタスク制御を容易に実現することができる。

【図面の簡単な説明】

【図１】本発明の実施の形態におけるリアルタイムタスク制御装置のブロック図

【図２】本発明の実施の形態におけるタスクスケジューリング動作を示すフロー図

【図３】本発明の実施の形態におけるタイマ値制御動作を説明するためのフロー図

【図４】本発明の実施の形態における実行タスクスイッチ動作を説明するためのフロー図

【図５】本発明の実施の形態において、タイムアウトに伴なって優先度の変更を行った場合の実行可能キューの状態例を示す図

【図６】本発明の実施の形態における各タスクのタスク実行時間およびタスク実行周期を示す図

【図７】本発明の実施の形態における各タスクの優先度の変化および実行タスクの変化を示す図

【図８】本発明の実施の形態における各タスクの最高タスク実行待ち時間およびタスク実行間待ち時間の合計を示す図

【図９】従来のスケジューリング方法を説明するための各タスクの優先度の変化及び実行タスクの変化を示す図

【図１０】従来例における各タスクの最高タスク実行待ち時間およびタスク実行間待ち時間の合計を示す図

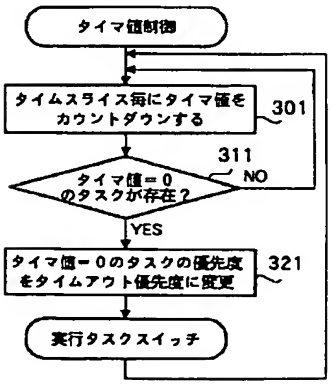
【図１１】他の従来例における各タスクの優先度の変化および実行タスクの変化を示す図

【図１２】他の従来例における各タスクの最高タスク実行待ち時間およびタスク実行間待ち時間の合計を示す図

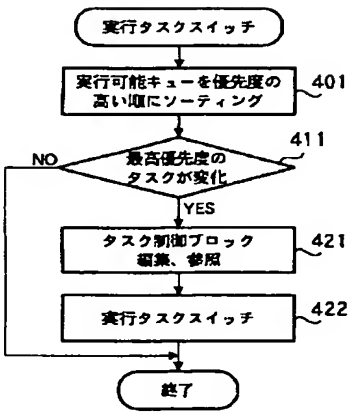
【符号の説明】

１００　リアルタイムタスク制御装置
１０１　タスク制御手段
１０２　タスクスケジューリング手段
１０３　タイマ値制御手段
１０４　タスク情報記憶手段
１０５　タスクスケジューリング手段
１１１　システムクロック
１２０　アプリケーションタスク群
１２１　アプリケーションタスク
１３０ａ～１３０ｃ　タスク制御ブロック
１４１　実行可能キュー

【図３】



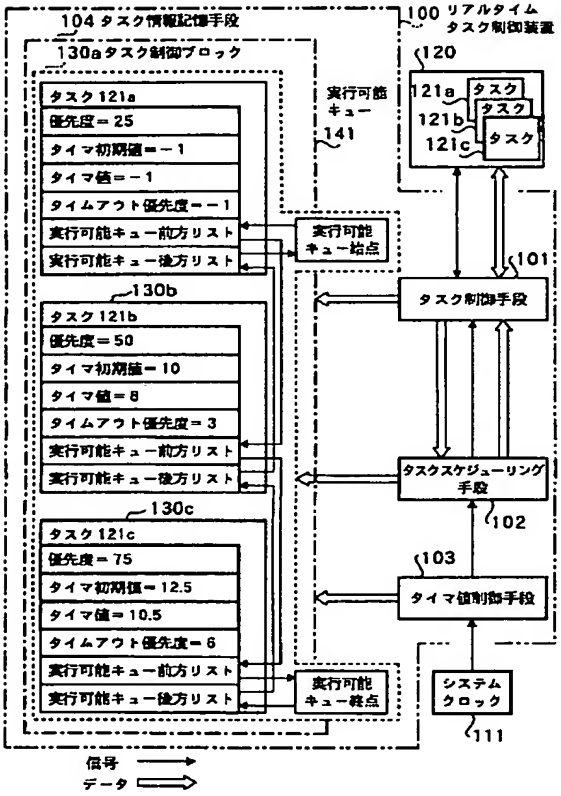
【図４】



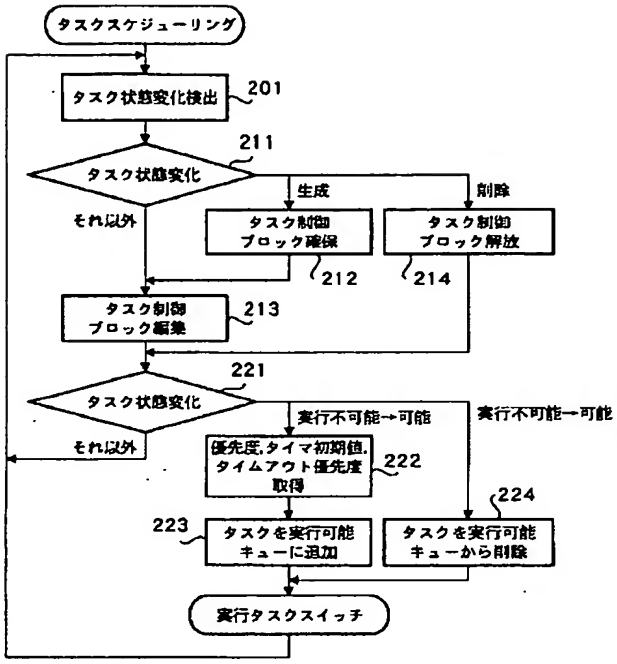
【図６】

タスク	タスク実行時間	タスク実行周期
タスク121a	15	30
タスク121b	5	20
タスク121c	1.25	40

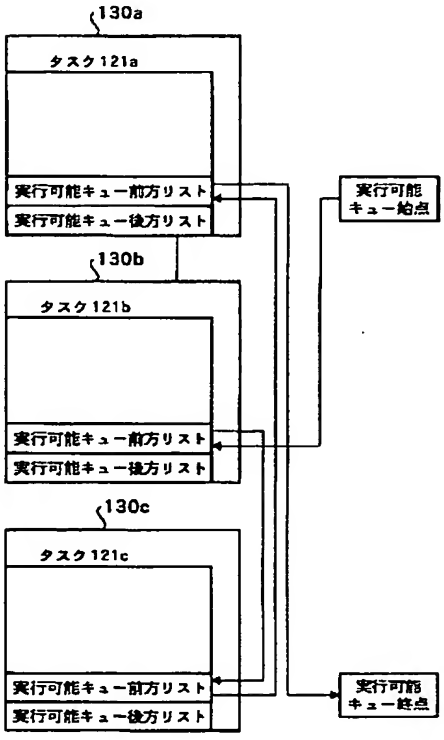
【図１】



【図２】



【図５】



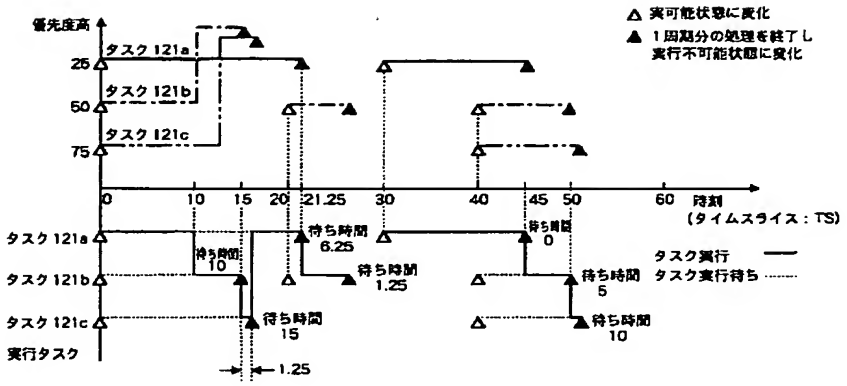
【図８】

タスク	最高タスク実行待ち時間 (TS)	タスク実行待ち時間合計 (TS)
タスク 121a	6.25	6.25
タスク 121b	10	16.25
タスク 121c	15	25
3つのタスクの合計	31.25	47.5

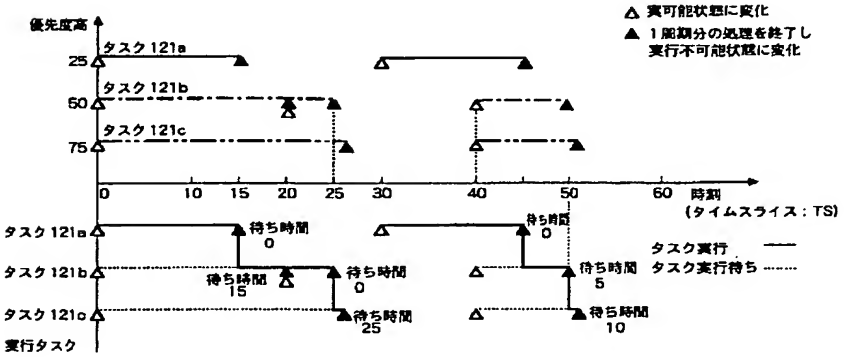
【図１０】

タスク	最高タスク実行待ち時間 (TS)	タスク実行待ち時間合計 (TS)
タスク 121a	0	0
タスク 121b	15	20
タスク 121c	25	35
3つのタスクの合計	40	55

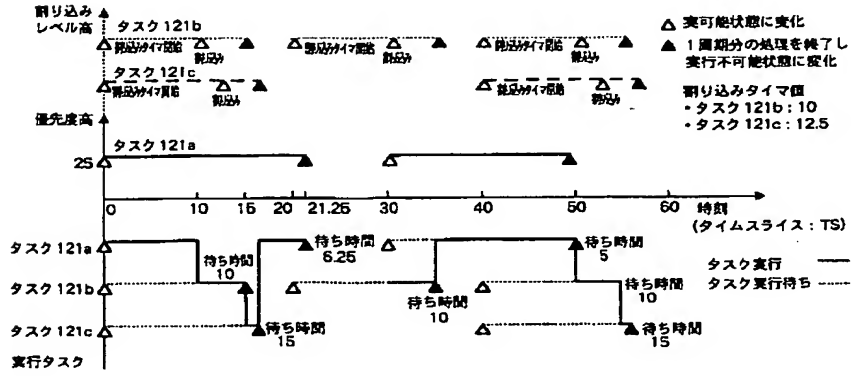
【図７】


優先度高
タスク 121a
25
タスク 121b
50
タスク 121c
75
△ 実可能状態に変化
▲ 1周期分の処理を終了し
実行不可能状態に変化
0 10 15 20 21.25 30 40 45 50 60 時刻
(タイムスライス：TS)
タスク 121a
タスク 121b
タスク 121c
実行タスク
待ち時間 10
待ち時間 6.25
待ち時間 1.25
待ち時間 15
待ち時間 0
待ち時間 5
待ち時間 10
1.25
タスク実行
タスク実行待ち

【図９】


優先度高
タスク 121a
25
タスク 121b
50
タスク 121c
75
△ 実可能状態に変化
▲ 1周期分の処理を終了し
実行不可能状態に変化
0 10 15 20 25 30 40 50 60 時刻
(タイムスライス：TS)
タスク 121a
タスク 121b
タスク 121c
実行タスク
待ち時間 0
待ち時間 15
待ち時間 0
待ち時間 25
待ち時間 0
待ち時間 5
待ち時間 10
タスク実行
タスク実行待ち

【図１１】


割り込み
レベル高
タスク 121b
タスク 121c
割込みタイマ起動
割込み
優先度高
タスク 121a
25
△ 実可能状態に変化
▲ 1周期分の処理を終了し
実行不可能状態に変化
割り込みタイマ値
・タスク 121b：10
・タスク 121c：12.5
0 10 15 20 21.25 30 40 50 60 時刻
(タイムスライス：TS)
タスク 121a
タスク 121b
タスク 121c
実行タスク
待ち時間 10
待ち時間 6.25
待ち時間 15
待ち時間 10
待ち時間 5
待ち時間 10
待ち時間 15
タスク実行
タスク実行待ち

【図１２】

タスク	最高タスク実行待ち時間 (TS)	タスク実行待ち時間合計 (TS)
タスク121a	6.25	11.25
タスク121b	10	30
タスク121c	15	30
3つのタスクの合計	31.25	71.25